AN IMPROVED CALIBRATION METHOD FOR DYNAMIC TRAFFIC ASSIGNMENT MODELS: CONSTRAINED EXTENDED KALMAN FILTER

Haizheng Zhang
Corresponding author
Massachusetts Institute of Technology
haizheng@mit.edu

Ravi Seshadri
Singapore-MIT Alliance for Research and Technology (SMART)
ravi@smart.mit.edu

Arun Prakash
Singapore-MIT Alliance for Research and Technology (SMART)
arun@smart.mit.edu

Francisco C. Pereira
Technical University of Denmark
camara@dtu.dk

Constantinos Antoniou
Technical University of Munich
c.antoniou@tum.de

Moshe Ben-Akiva
Massachusetts Institute of Technology
mba@mit.edu

Word Count: 5512 words + 5 figure(s) + 2 table(s) = 7262words

1 ABSTRACT

The calibration of dynamic traffic assignment (DTA) models involves the estimation of model parameters so as to best replicate real world measurements. A good model calibration is essential to accurately estimate and predict traffic states, which are crucial for traffic management applications to alleviate congestion.

A widely used solution approach to calibrate simulation-based DTA models is the Extended Kalman Filter (EKF). The EKF assumes that the DTA model parameters are unconstrained, although they are in fact constrained – for instance, OD flows are non-negative. This assumption is typically not problematic for small and medium scale networks where the EKF has been successfully applied in the past. However, in the case of large scale networks (which typically contain a large number of OD pairs with small magnitudes of flow), the estimates may violate the constraints severely. In consequence, simply truncating the infeasible estimates may result in the divergence of EKF, leading to extremely poor state estimates and predictions. To address this issue, we present a Constrained EKF (CEKF) approach, which imposes constraints on the posterior distribution of the state estimators to obtain the maximum a posteriori (MAP) estimates that are feasible. The feasible MAP estimates are obtained using a heuristic followed by the coordinate descent method. The procedure determines the optimum and was found to be computationally faster by 31.5% over coordinate descent and by 94.9% over interior point method.

Experiments on the Singapore expressway network indicate that the CEKF significantly improves model accuracy and outperforms both the traditional EKF (by up to 78.17%) and Generalized Least Squares (by up to 17.13%) approaches in state estimation and prediction.

2 INTRODUCTION

Traffic management policies and strategies are essential in controlling congestion and mitigating its negative impacts. In order to be effective, these measures significantly depend on accurate estimates and predictions of traffic states. Dynamic Traffic Assignment (DTA) systems are effective in evaluating the current and future performance of transportation facilities [1], as they model the complex supply and demand interactions [2] (refer [3] for a more detailed discussion of simulation based DTA systems). However, for DTA systems to be effective, they need to be properly calibrated.

The Extended Kalman Filter (EKF) is one of the classical approaches used to calibrate DTA systems. The EKF assumes that the state vector (which represents model parameters to be calibrated) is unconstrained, whereas in fact, some parameters like OD flows violate this assumption because they are non-negative. In the past, for simpler networks, the EKF has shown satisfactory performance despite this assumption [4]. However, on larger networks − like the Singapore expressway network considered in this study − the origin-destination flow estimates from EKF tend to intermittently violate the non-negativity constraints. Truncating the infeasible OD flow estimates to zero can result in erroneous estimates and can artificially induce extra demand. To overcome this issue, we propose a Constrained Extended Kalman Filter (CEKF) that explicitly models the constraints on the parameters. The constraints are imposed by determining the maximum a posteriori (MAP) estimates subject to the constraints.

The contributions of this work to the existing literature are as follows. First, we adapt the CEKF to the DTA context and analyze its performance on a large real-world network, considering both state estimation and state prediction. Second, we apply a procedure that iteratively adds equality constraints followed by the coordinate descent method to obtain the MAP estimates. Third, we demonstrate that CEKF improves over the EKF significantly in both state estimation (by up to 78.17%) and state prediction (by up to 76.38%). Results show that it also outperforms the GLS approach in both estimation and prediction (by up to 17.13%).

3 LITERATURE REVIEW

The calibration of simulation-based DTA models has received considerable attention in the literature in primarily two contexts, offline and online. The offline calibration problem typically involves estimating historical values for simulation parameters to ensure that the simulator can closely replicate average traffic conditions for a given network [5]. In contrast, the online problem involves updating the historical parameters in real-time based on prevailing traffic conditions [4]. For a detailed review of online and offline calibration of DTA systems refer [4] and [5] respectively.

The existing approaches to both the offline and online problems are based on either optimization formulations or state space formulations. The former involve primarily generalized least squares (GLS) approaches such as that of [6, 7] for the dynamic OD estimation problem. Although the GLS approach explicitly handles non-negativity constraints on the OD flows it assumes a linear mapping between the measurements and parameters making it difficult to incorporate supply side parameters. Balakrishna et al. [8] propose a more generic solution method for the offline problem based on the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm to simultaneously calibrate demand and supply parameters that can incorporate any type of measurement.

The second class of approaches are Kalman filtering (KF) based techniques which have also been applied to both the offline and online versions of the calibration problem. Ashok [9] proposed a Kalman filtering method where the state variables are deviations of OD flows from historical values (rather than the OD flows themselves). Building on this, Ashok and Ben-Akiva [10] proposed a modified approach that explicitly accounts for stochasticity in the assignment matrix (which maps OD flows to traffic counts). The Kalman filter has also been applied to the real time estimation of OD flows by Zhou and Mahmassani [11] where the transition equation is a polynomial trend filter designed to capture historical trends and structural deviations.

Antoniou et al. [12] further extended the work of Ashok [9] to jointly estimate demand and supply parameters of dynamic traffic assignment (DTA) systems. The authors propose three extensions of the Kalman filtering algorithm including the extended Kalman filter (EKF), the limiting EKF (LimEKF), and the unscented Kalman filter. Numerical experiments on a small network indicated that the LimEKF yields an accuracy that is comparable to the other algorithms but with vastly superior computational performance.

The KF based algorithms are attractive for both the offline and online calibration problems as they can handle any calibration parameters and any type of measurement data [13]. However, one limitation of these approaches is the assumption that the state variables are unconstrained. Although this is not an issue for smaller networks, the violation of constraints (such as the non-negativity of OD flows) can be severe for large networks (as numerical experiments in Section 7 suggest) and naive truncation procedures can lead to inaccurate estimates and predictions.

The problem of modeling constraints within a Kalman Filter has received attention in other domains (see for instance, [14, 15]), but has largely been ignored in the context of KF methods for DTA calibration. Moreover, the performance of the KF algorithms has not been systematically tested on large scale networks. This study aims to address these issues by adapting a constrained EKF model to the DTA calibration problem and testing the performance of this algorithm on the Singapore expressway network.

4 STATE SPACE FORMULATION

The calibration problem for DTA systems lends itself to a state space formulation which is a classic approach to model dynamic systems. A state space model is defined by a state vector that succinctly captures the state of the system through a set of variables, a transition equation that captures the evolution of the state vector over time and a measurement equation that maps the state vector to the measurements. We denote the state vector by $\boldsymbol{x}_h$ which consists of selected parameters of the DTA model to be calibrated (for a given time interval h) and typically includes the time-dependent OD flows, segment-based supply parameters and route choice parameters. The time-dependent measurements from the real world are denoted by $\boldsymbol{M}_h$. Thus, the state space model can be written as:

- Transition equation

$$\boldsymbol{x}_h = \boldsymbol{f}(\boldsymbol{x}_{h-1}, ..., \boldsymbol{x}_{h-p}) + \boldsymbol{w}_h \tag{1}$$

- Measurement equation

$$\boldsymbol{M}_h = \boldsymbol{g}(\boldsymbol{x}_h, ..., \boldsymbol{x}_{h-q+1}) + \boldsymbol{v}_h \tag{2}$$

where, h is the time interval index, $h \in \mathcal{H} = \{1, 2, ..., H\}$; p is the number of previous states that are believed to be related to $\boldsymbol{x}_h$; q is the number of previous states that affect the current measurement $\boldsymbol{M}_h$; $\boldsymbol{f}$ represents the relationship between state vectors of different intervals (or temporal dependence) ; $\boldsymbol{g}$ is the simulation model (Dynamic traffic assignment system in our context) which maps the state vector $\boldsymbol{x}_h$ to the measurement vector $\boldsymbol{M}_h$; $\boldsymbol{w}_h$ and $\boldsymbol{v}_h$ are random errors.

The transition equation is typically modeled as an autoregressive process [10, 12] and hence, we have,

$$\boldsymbol{x}_h = \sum_{k=h-p}^{h-1} \boldsymbol{F}_h^k \boldsymbol{x}_k + \boldsymbol{w}_h \tag{3}$$

where, p and q are the same as in Equations (1) and (2); matrix $\boldsymbol{F}_h^k$ is a matrix of autoregressive coefficients that relate the state vector in time interval k to the state vector in the current time interval h.

The Idea of Deviations

Although the autoregressive process in Equation (3) captures temporal dependencies between the state variables, it does not represent structural information about trip patterns. Along the lines of [9], the state space model can be formulated in terms of deviations from historical values to better capture structural relationships (for instance, spatial and temporal distribution of activities and characteristics of the transportation system). The use of deviations is also more amenable to the application of Kalman fliter based solution approaches which assume a Gaussian distribution for the state vector. Thus, the deviations are defined as:

$$\partial \boldsymbol{x}_h = \boldsymbol{x}_h - \boldsymbol{x}_h^H \tag{4}$$

$$\partial \boldsymbol{M}_h = \boldsymbol{M}_h - \boldsymbol{M}_h^H \tag{5}$$

where, $\partial \boldsymbol{x}_h$ and $\partial \boldsymbol{M}_h$ are the deviations for state vector $\boldsymbol{x}_h$ and measurement vector $\boldsymbol{M}_h$. $\boldsymbol{x}_h^H$ and $\boldsymbol{M}_h^H$ are the corresponding historical values. The transition and measurement equations can now be written in terms of deviations as,

$$\partial \boldsymbol{x}_h = \sum_{k=h-1}^{h-p} \boldsymbol{F}_h^k \partial \boldsymbol{x}_k + \boldsymbol{w}_h \tag{6}$$

$$\partial \boldsymbol{M}_h = \boldsymbol{g}(\partial \boldsymbol{x}_h + \boldsymbol{x}_h^H, \ldots, \partial \boldsymbol{x}_{h-q+1} + \boldsymbol{x}_{h-q+1}^H) - \boldsymbol{M}_h^H + \boldsymbol{v}_h \tag{7}$$

where, p is the same as in Equation (1); matrix $\boldsymbol{F}_h^k$ is a matrix of autoregressive coefficients that relates the deviation of the DTA parameters from historical values in time interval k to the deviations in the current time interval h. It is noted that state vector is a term in state space formulation and used in Kalman filters. In the rest of this paper, deviations are used as state vector.

5 EXTENDED KALMAN FILTER (EKF)

This section briefly describes approaches to solve the state space model formulated in the last section. The classical Kalman Filter (KF) which is the optimal minimum mean square error

(MMSE) estimator for linear state-space models [4] is first discussed followed by a brief outline of the Extended Kalman Filter (EKF) which handles the non-linearity in the measurement equation (Equation (7)). Note that for the application of the KF based methods, we impose an additional assumption we useon the error terms $\boldsymbol{w}_h$ and $\boldsymbol{v}_h$ (Equations (6) and (7)), namely that they are zero mean Gaussian variables.

The main steps of the KF algorithm are as follows. Assuming we have the optimal estimates of the previous time step $h-1$: $\partial\hat{\boldsymbol{x}}_{h-1|h-1}$ and $\boldsymbol{P}_{h-1|h-1}$ (covariance matrix of the state vector), a *time update* phase makes a prediction of the state $\partial\hat{\boldsymbol{x}}_{h|h-1}$ and its covariance matrix $\boldsymbol{P}_{h|h-1}$ for the next time interval. These are termed the prior estimates. The measurement update phase then incorporates the new information about the measurement vector and uses it to correct the prediction of the state made during the time update. The updated estimates $\partial\hat{\boldsymbol{x}}_{h|h}$ and $\boldsymbol{P}_{h|h}$ are called posterior estimates. For a detailed description of the KF algorithm refer [4].

The original KF algorithm applies to linear systems, i.e. it assumes linearity of both the transition and measurement equations. The most straightforward extension of the KF methodology to handle non-linearity is the Extended Kalman Filter (EKF) which employs a linearization of the non-linear relationship (measurement equation in our case) using a first order Taylor series expansion. Thus, the measurement equation is approximated by,

$$\partial\boldsymbol{M}_h = \sum_{k=h}^{h-q+1} \boldsymbol{H}_h^k \partial\boldsymbol{x}_k + \boldsymbol{v}_h \tag{8}$$

where, now the $\boldsymbol{H}_h^k$ represents the linear relation between $\partial\boldsymbol{M}_h$ and $\partial\boldsymbol{x}_k$. Since the measurement equation involves the DTA model, it does not have an analytical representation and hence, in order to perform the linearization it is necessary to use *numerical derivatives*. We use a standard central finite difference method to compute the gradient of $\boldsymbol{g}_h$.

Limitation of the EKF

As noted previously, the standard KF, EKF algorithms assume that the state vector is unconstrained and the error terms $\boldsymbol{w}_h$ and $\boldsymbol{v}_h$ (Equations (8) and (6)) are Gaussian. However, the parameters of DTA models (that are to be calibrated) are in fact constrained. For instance, OD flows are necessarily non-negative, and hence, if the state vector consists of only the time-dependent OD flows, we must have,

$$\boldsymbol{x}_h \geq \boldsymbol{0} \implies \partial\boldsymbol{x}_h + \boldsymbol{x}_h^H \geq \boldsymbol{0} \tag{9}$$

Thus, from Equations Equations (8) and (9), and given that $\boldsymbol{H}_h^k$ contains non-negative elements (when $\boldsymbol{x}_h$ consists of OD flows and $\boldsymbol{M}_h$ consists of sensor counts) we must have,

$$\boldsymbol{v}_h \leq \partial\boldsymbol{M}_h + \sum_{k=h}^{h-q+1} \boldsymbol{H}_h^k \boldsymbol{x}_k^H \tag{10}$$

In other words, due to the constraints on the state vector, the error term in the measurement equation is also constrained such that the probability density for some values are strictly zero. Thus, strictly speaking, modeling it with a Gaussian distribution is not correct.

In practice, when we have constraints on the state vector, a simple way to impose them is to project the estimated state vector onto the feasible region. When the constraints are in the form of lower and upper bounds, we can simply project each element of the state vector onto its corresponding feasible region. We refer to this element-wise projection as *truncation*. Although efficient, this procedure is not necessarily correct, because estimators of different dimensions are correlated. Truncating one variable while keeping others intact disregards its relationship with other variables.

In DTA calibration, this truncation is consequential particularly when the true values of the OD flows are zero or close to zero, and the estimated variance is large. In this case, the Kalman filter tends to give estimates with noise around the true value. For the OD pairs with 0 as true values, the estimates will be either positive or negative. Then, due to the truncation, the negative values will be set to zero leading to an overestimation of total demand. Since this overestimation happens for each interval, the error would accumulate leading to poor state estimates. To address this issue, in the next section, we present a modification of the EKF that explicitly handles constraints on the state we usevector.

6 CONSTRAINED EXTENDED KALMAN FILTER

This section discusses the proposed Constrained Extended Kalman Filter (CEKF) method. The intuition and theoretical basis are first presented followed by a description of the solution algorithm.

Optimization Formulation for Constrained Kalman Filter Estimates

In this section, for ease of presentation, we use $\boldsymbol{x}_h$ to denote the state vector which is understood to be the deviations from historical values. The EKF estimates $\hat{\boldsymbol{x}}_{h|h}$ at time step h are essentially the maximum *a posteriori* (MAP) estimates, which are obtained from the measurements and the prior distribution (based on the transition equation). The posterior Gaussian distribution of the state estimate is given by:

$$f_{X_{h|h}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{P}_{h|h}|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}}_{h|h})^\top \boldsymbol{P}_{h|h}{}^{-1}(\boldsymbol{x} - \hat{\boldsymbol{x}}_{h|h})\right\} \tag{11}$$

where, n is the dimension of vector $\boldsymbol{x}$, and $\boldsymbol{P}_{h|h}$ is the posterior covariance matrix.

For example, consider the case where we have two state variables (x, y) and assume that the posterior distribution is given by $(x, y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where,

$$\boldsymbol{\mu} = [0.5, -1]^\top$$
$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$$

A contour plot of the posterior probability density function (PDF) is shown in Figure 1. We can see that the "cross" is the center of the PDF, which is the MAP estimate for the unconstrained EKF. Now assume that the state variables are non-negative. When we directly impose the constraints $x \geq 0, y \geq 0$, i.e set $y = 0$, we obtain the "circle" point. But in terms of maximizing *a posteriori* probability density under the constraints, the "circle" point is clearly sub-optimal. The true MAP estimate is the "asterisk" point.

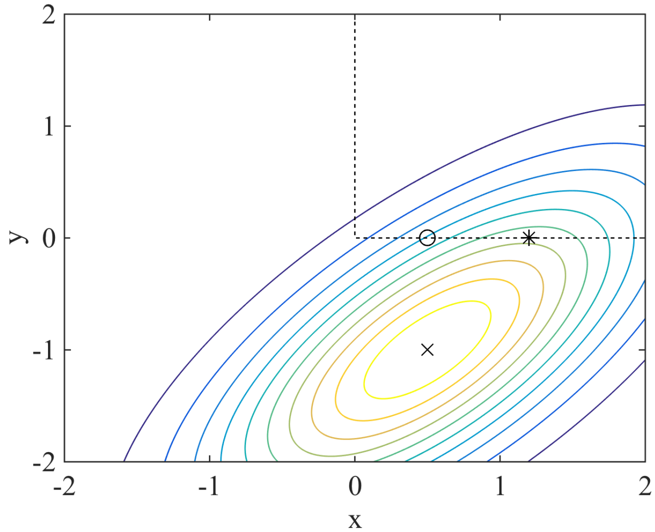


FIGURE 1 : 2-D Posterior PDF Contour and Different Estimators

Formally, the problem of computing the MAP estimate under the constraints is termed the Kalman filter with state inequality constraints and is discussed at length in [14, 15]. It can be formulated as a quadratic program subject to linear inequality constraints:

$$\max_{\boldsymbol{x}} f_{X_{h|h}}(\boldsymbol{x}) \Leftrightarrow \min_{\boldsymbol{x}} (\boldsymbol{x} - \hat{\boldsymbol{x}}_{h|h})^{\top} \boldsymbol{P}_{h|h}{}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{x}}_{h|h}) \tag{12}$$

$$\text{s.t. } \boldsymbol{Dx} \leq \boldsymbol{d} \tag{13}$$

where, $\boldsymbol{D}$ is a known $s \times n$ constant matrix, s is the number of constraints, n is the number of state variables, and $s \leq n$; Further, $\boldsymbol{D}$ is assumed to be a full rank matrix, i.e. its rank is s. If the rank of $\boldsymbol{D}$ is less than s, we can always drop the redundant constraints to make it full rank.

An Efficient Near-Optimal Algorithm for EKF with Bound Constraints in DTA Calibration

In the context of DTA calibration, the constraints are usually in the form of bounds on model parameters. For instance, in OD estimation (where the state vector $\boldsymbol{x}$ consists of OD flows; for brevity, we drop the time interval subscript), we have $\boldsymbol{x} \geq 0$. Similarly, supply parameters $\boldsymbol{s}$, could have both upper bounds and lower bounds, i.e. $\boldsymbol{s}^{lb} \leq \boldsymbol{s} \leq \boldsymbol{s}^{ub}$. Thus, for the DTA calibration problem, we have the following optimization formulation after each measurement update of the EKF:

$$\min_{\boldsymbol{x}} (\boldsymbol{x} - \overline{\overline{\boldsymbol{x}}})^{\top} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \overline{\overline{\boldsymbol{x}}}) \tag{14}$$

$$\text{s.t. } \boldsymbol{x}^{lb} \leq \boldsymbol{x} \leq \boldsymbol{x}^{ub}$$

where, $\overline{\overline{\boldsymbol{x}}} = \hat{\boldsymbol{x}}_{h|h}, \boldsymbol{\Sigma} = \boldsymbol{P}_{h|h}$, $\boldsymbol{x}^{lb}$ and $\boldsymbol{x}^{ub}$ are the lower and upper bounds for the state vector $\boldsymbol{x}$.

An intuitive method of solving the above optimization problem is based on the concept of truncation described earlier. For simplicity, assume there exists a lower bound $\boldsymbol{x}^{lb}$ on $\boldsymbol{x}$, but no upper bound. When we truncate $\boldsymbol{x}$, we set the elements that violate the lower bounds to the corresponding elements in $\boldsymbol{x}^{lb}$. In essence, we are introducing equality constraints to the optimization problem. Let $\mathcal{A}$ denote the set of indices of the state variables on which truncation is performed and let $\boldsymbol{x}_{\mathcal{A}}$ denote the corresponding vector. In addition, let $\mathcal{A}^c$ denote the complement of the set $\mathcal{A}$ and let $\boldsymbol{x}_{\mathcal{A}^c}$ denote the corresponding vector. In order to compute the MAP subject to the constraints, we need to maximize the following conditional PDF:

$$\max_{\boldsymbol{x}_{\mathcal{A}^c}} f_X\left(\boldsymbol{x}_{\mathcal{A}^c}|\boldsymbol{x}_{\mathcal{A}}=(\boldsymbol{x}^{lb})_{\mathcal{A}}\right) \tag{15}$$

Maximizing the conditional probability in Equation (15) is equivalent to maximizing the joint probability $f_X\left(\boldsymbol{x}_{\mathcal{A}^c},\boldsymbol{x}_{\mathcal{A}}=(\boldsymbol{x}^{lb})_{\mathcal{A}}\right)$, since:

$$f_X\left(\boldsymbol{x}_{\mathcal{A}^c}|\boldsymbol{x}_{\mathcal{A}}=(\boldsymbol{x}^{lb})_{\mathcal{A}}\right)=\frac{f_X\left(\boldsymbol{x}_{\mathcal{A}^c},\boldsymbol{x}_{\mathcal{A}}=(\boldsymbol{x}^{lb})_{\mathcal{A}}\right)}{f_X\left(\boldsymbol{x}_{\mathcal{A}}=(\boldsymbol{x}^{lb})_{\mathcal{A}}\right)}$$

and the denominator is constant for a given $\overline{\overline{\boldsymbol{x}}}$ and $\boldsymbol{\Sigma}$. Thus, we have,

$$\max_{\boldsymbol{x}_{\mathcal{A}^c}} f_X\left(\boldsymbol{x}_{\mathcal{A}^c},\boldsymbol{x}_{\mathcal{A}}=(\boldsymbol{x}^{lb})_{\mathcal{A}}\right)\Leftrightarrow \min_{\boldsymbol{x}_{\mathcal{A}^c}}\,(\boldsymbol{x}-\overline{\overline{\boldsymbol{x}}})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\overline{\overline{\boldsymbol{x}}})\Big|\,\boldsymbol{x}_{\mathcal{A}}=(\boldsymbol{x}^{lb})_{\mathcal{A}} \tag{16}$$

With some algebraic manipulation (refer [16]), it can be shown that the solution to the optimization problem in Equation (16) is given by,

$$\boldsymbol{x}_{\mathcal{A}^c}=\overline{\overline{\boldsymbol{x}}}_{\mathcal{A}^c}+\boldsymbol{\Sigma}_{\mathcal{A}^c,\mathcal{A}}(\boldsymbol{\Sigma}_{\mathcal{A},\mathcal{A}})^{-1}\left(\boldsymbol{x}_{\mathcal{A}}-\overline{\overline{\boldsymbol{x}}}_{\mathcal{A}}\right) \tag{17}$$

$$\boldsymbol{x}_{\mathcal{A}}=(\boldsymbol{x}^{lb})_{\mathcal{A}} \tag{18}$$

where $\boldsymbol{\Sigma}_{\mathcal{A}^c,\mathcal{A}}$ is the covariance matrix between $\boldsymbol{x}_{\mathcal{A}^c}$ and $\boldsymbol{x}_{\mathcal{A}}$, and $\boldsymbol{\Sigma}_{\mathcal{A},\mathcal{A}}$ is the covariance matrix of $\boldsymbol{x}_{\mathcal{A}}$.

In a similar fashion, for the general case in Equation (14), when the MAP estimates of the unconstrained EKF violate the bounds (whose indices are in Set $\mathcal{A}$), we can project them back to the boundary, and then obtain the conditional MAP with $\boldsymbol{x}_{\mathcal{A}}$ fixed to the bounds, according to Equation (17). Note that this conditional MAP solution is not guaranteed to satisfy the bounds for $\boldsymbol{x}_{\mathcal{A}^c}$. Thus, this procedure needs to be performed iteratively, where the indices of state variables whose bounds are violated from set $\mathcal{A}^c$ are added to set $\mathcal{A}$. We then re-estimate the conditional MAP, until all elements whose indices are in set $\mathcal{A}^c$ are in the feasible region. The near-optimal algorithm is referred to as Algorithm 1, where $\boldsymbol{x}_{\mathcal{A}}=[\boldsymbol{x}_{\mathcal{A}(1)},...,\boldsymbol{x}_{\mathcal{A}(|\mathcal{A}|)}]^\top$, $\mathcal{A}(j)$ is the j-th element in Set $\mathcal{A}$, $|\mathcal{A}|$ is the cardinality of Set $\mathcal{A}$; Similarly, $\boldsymbol{\Sigma}_{\mathcal{A}^c,\mathcal{A}}=[\boldsymbol{\Sigma}_{i,j}]\big|_{i,j\in\mathcal{A}^c\times\mathcal{A}}$.

Based on our experiments, this algorithm gives an objective function (Equation (14)) value less than 0.1% worse than the true optimal (obtained by solving the original quadratic programming problem exactly), but is more efficient. This is discussed in more detail in the next subsection.

Algorithm 1 EKF with Iterative Addition of Equality Constraints

Run EKF and obtain state estimate $\bar{\bar{\boldsymbol{x}}}$ and variance estimate $\boldsymbol{\Sigma}$, n is the dimension of $\bar{\bar{\boldsymbol{x}}}$
Initialize

$$\mathcal{I} \leftarrow \varnothing$$
$$\mathcal{A} \leftarrow \varnothing$$
$$\boldsymbol{x} \leftarrow \bar{\bar{\boldsymbol{x}}}$$

do

if $\mathcal{I} \neq \varnothing$ **then**

Adjust invalid state elements

$$\boldsymbol{x}_{\mathcal{I}_{lb}} \leftarrow \boldsymbol{x}^{lb}_{\mathcal{I}_{lb}} \tag{19}$$
$$\boldsymbol{x}_{\mathcal{I}_{ub}} \leftarrow \boldsymbol{x}^{ub}_{\mathcal{I}_{ub}} \tag{20}$$

Find conditional MAP estimates

$$\mathcal{A} \leftarrow \mathcal{A} \bigcup \mathcal{I} \tag{21}$$
$$\mathcal{A}^c \leftarrow \{1, 2, \ldots, n\} \setminus \mathcal{A} \tag{22}$$
$$\boldsymbol{x}_{\mathcal{A}^c} = \bar{\bar{\boldsymbol{x}}}_{\mathcal{A}^c} + \boldsymbol{\Sigma}_{\mathcal{A}^c,\mathcal{A}}(\boldsymbol{\Sigma}_{\mathcal{A},\mathcal{A}})^{-1}\left(\boldsymbol{x}_{\mathcal{A}} - \bar{\bar{\boldsymbol{x}}}_{\mathcal{A}}\right) \tag{23}$$

end if

$$\mathcal{I}_{lb} \leftarrow \varnothing$$
$$\mathcal{I}_{ub} \leftarrow \varnothing$$

Identify invalid state indices
for $j = 1$ to n **do**
if $\boldsymbol{x}_j < \boldsymbol{x}^{lb}_j$ **then** $\mathcal{I}_{lb} \leftarrow \mathcal{I}_{lb} \bigcup \{j\}$
else if $\boldsymbol{x}_j > \boldsymbol{x}^{ub}_j$ **then** $\mathcal{I}_{ub} \leftarrow \mathcal{I}_{ub} \bigcup \{j\}$
end if
end for

$$\mathcal{I} \leftarrow \mathcal{I}_{lb} \bigcup \mathcal{I}_{ub} \tag{24}$$

while $\mathcal{I} \neq \varnothing$

Coordinate Descent Algorithm with Near-Optimal Initialization
When we are interested in computing the true optimum solution to the optimization problem in Equation (14), Algorithm 1 can serve to provide an initial estimate or a starting point for solution of the quadratic program. Since the DTA calibration problem involves independent constraints for each element, a coordinate descent method can be applied to solve the quadratic programming problem. The coordinate descent algorithm is referred to as Algorithm 2.

Algorithm 2 Coordinate Descent

Initialize

$$\begin{aligned} \boldsymbol{x} &\leftarrow \boldsymbol{x}_0 \\ \epsilon &\leftarrow 0.001 \\ \boldsymbol{Q} &\leftarrow \boldsymbol{\Sigma}^{-1} \\ \boldsymbol{b} &\leftarrow -\boldsymbol{\Sigma}^{-1}\overline{\overline{\boldsymbol{x}}} \\ Obj_{this} &\leftarrow (\boldsymbol{x}-\overline{\overline{\boldsymbol{x}}})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\overline{\overline{\boldsymbol{x}}}) \end{aligned}$$

do

for $j = 1$ to n **do**

$$\boldsymbol{x}_j = \boldsymbol{x}_j - \frac{1}{\boldsymbol{Q}_{j,j}}\left(\boldsymbol{Q}_{j,1:n}\boldsymbol{x} + \boldsymbol{b}_j\right) \tag{25}$$

$$\boldsymbol{x}_j \leftarrow \max(\boldsymbol{x}_j, \boldsymbol{x}_j^{lb}) \tag{26}$$

$$\boldsymbol{x}_j \leftarrow \min(\boldsymbol{x}_j, \boldsymbol{x}_j^{ub}) \tag{27}$$

end for

$$Obj_{last} \leftarrow Obj_{this} \tag{28}$$

$$Obj_{this} \leftarrow (\boldsymbol{x}-\overline{\overline{\boldsymbol{x}}})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\overline{\overline{\boldsymbol{x}}}) \tag{29}$$

while $Obj_{last} - Obj_{this} > \epsilon$

Several remarks are in order regarding the coordinate descent algorithm. First, the step size in each update is fixed to $\frac{1}{\boldsymbol{Q}_{j,j}}$. Since the objective function is quadratic, an update using this step size will yield the optimal solution for $\boldsymbol{x}_j$, when other dimensions are fixed. Second, this algorithm is computationally inexpensive, because there are no matrix multiplications in Equation (25). Last but not least, in the specific context of OD estimation, other objective functions could be used as the stopping rule. For instance, a distance measurement (like L_1 norm) between the current and the last estimated state vector could be used as the objective function. When the improvement of the objective function is less than ϵ, the algorithm terminates.

The performance of Algorithm 1, Algorithm 2, Algorithm 2 with initial solution obtained from Algorithm 1 (termed Algorithm 1+2), and an interior point algorithm to directly solve the quadratic program (implemented using the *quadprog* function in MATLAB) are compared using 5 arbitrary time intervals from the simulation experiments described in Section 7. In order to reach

TABLE 1 : Objective Function Value and Computation Time of 5 Examples in Calibration

	#	Truncate	Alg1	Alg2	Alg1+Alg2	quadprog
Objective Function Value	1	337.33710	297.78574	297.78572	297.78572	297.78572
	2	379.27500	319.66499	319.59357	319.59357	319.59357
	3	244.22211	178.54625	178.54420	178.54420	178.54420
	4	444.13678	346.88061	346.88060	346.88060	346.88060
	5	635.38129	448.67091	448.66981	448.66981	448.66981
Computation Time (milliseconds)	1	0.1	70.8	853.4	470.5	10613.3
	2	0.1	75.3	1138.7	650.0	10002.9
	3	0.1	137.9	832.3	485.4	9537.3
	4	0.1	167.2	1003.4	524.2	14143.3
	5	0.1	605.0	2950.2	2514.2	47432.4

the same precision, the convergence criterion of Algorithm 2, Algorithm 1+2 and *quadprog* is all set to $\|\boldsymbol{x}_i - \boldsymbol{x}_{i-1}\| < 10^{-3}$, where $\boldsymbol{x}_i$ is the solution obtained from current iteration.

The results indicate that in all tests cases the objective function value using the naive truncation procedure is significantly worse than all the four aforementioned solution methods which yield similar objective function values. In terms of computational time, clearly Algorithm 1 is substantially faster than the other procedures and Algorithm 1 + 2 significantly outperforms Algorithm 2. However, given that optimality is not guaranteed for Algorithm 1, we choose Algorithm 1 + 2 (over algorithm 2, and *quadprog*) for all the subsequent experiments in view of its superior computational performance. It is noted that although the quadratic programming algorithms have polynomial complexity, performance may still deteriorate significantly for higher dimensions. In such cases, dimensionality reduction procedures (e.g. PCA, factor analysis) may be used to maintain computational tractability.

7 APPLICATION ON SINGAPORE EXPRESSWAY NETWORK

In this section, we test the performance of the EKF, CEKF, and a Generalized Least Squares method (GLS) [10] on the Singapore expressway network. We adopt an open loop framework where the DTA system interacts with a microsimulator that emulates the real world.

Simulation Setup

The experiment was conducted on Singapore expressway network (dark orange links in Figure 2) using DynaMIT [3] as the real-time DTA system. The network consists of 939 nodes and 1157 links, 1623 origin-destination (OD) pairs with time-dependent flows, and 650 segment specific sensors. As per the notation in Equations (6) and (7), $\partial \boldsymbol{x}_h$ is the deviation in OD flow for interval h, and $\partial \boldsymbol{M}_h$ is the deviation between real-time sensor counts and historical sensor counts.

For the experiment, the calibration variables are the time-dependent origin-destination flows. The supply parameters and behavioral parameters were obtained from a prior offline calibration using the W-SPSA algorithm [17]. The simulation period was from 17:00 to 21:30, which includes the evening peak. The chosen estimation interval was 5 minutes and the prediction interval was 15 minutes. Note that, as the experiment is in the online setting, parameters are calibrated interval-wise sequentially.

For the simulation setup, we used an open-loop framework, wherein a traffic microsimulator

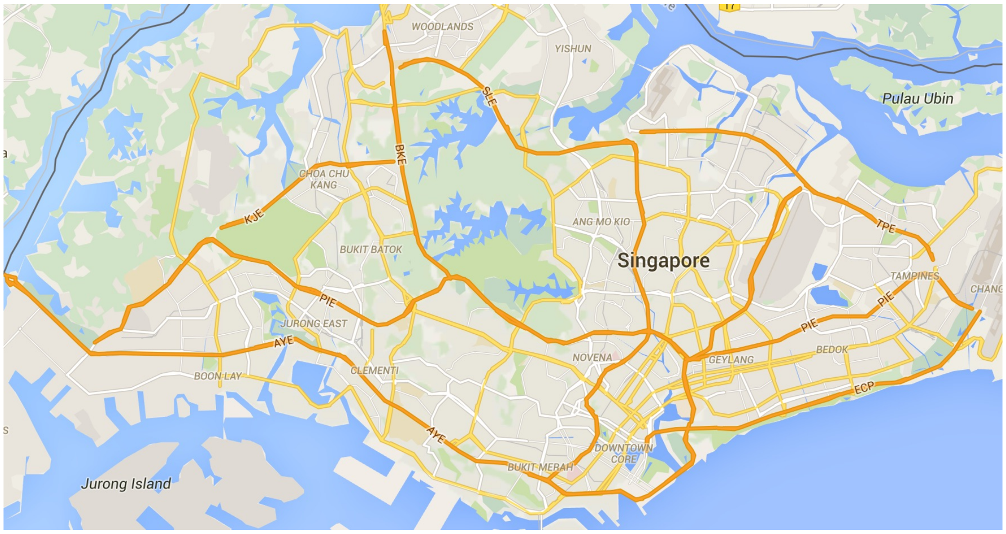


FIGURE 2 : Singapore Road Network (*source: Google Maps, 2016*)

(MITSIMLab [18]) emulates the real-world, generates the surveillance data and feeds it to the DTA system. Then the DTA system utilizes those data and performs calibration.

Demand generation

To setup the open-loop environment, MITSIMlab was calibrated against the real-world sensor data using the W-SPSA algorithm [17]. However, as the calibrated time-dependent demand displayed high fluctuation between consecutive intervals, it was smoothed using a Gaussian kernel with a bandwidth h of 10 minutes. The resulting demand is more representative of the real-world and is termed the "actual" demand, which is an input to MITSIMLab.

The historical demand for DynaMIT was generated by perturbing the "actual" demand in MITSIMLab. The rationale being that the historical demand is generally a reasonable approximation of the true demand. The historical demand in DynaMIT was accordingly constructed as follows:

$$x_{h,i}^{H} = (0.75 + 0.15z) \times x_{h,i}^{true} \tag{30}$$

$$z \sim N(\mu = 0, \sigma^2 = \frac{1}{9}) \tag{31}$$

where h is the time interval, i is the index of the OD pair and $x_{h,i}^{true}$ indicates the actual demand for i-th OD pair at time h. z is a zero mean Gaussian random number with $\sigma = \frac{1}{3}$ so that statistically 99.7% of the coefficients (multipliers in Equation (30)) are between 0.6 to 0.9. The randomness ensures that historical demand has a different pattern from the actual demand, as the true demand is generally not known. The historical demand was underestimated to avoid the DTA system from being oversaturated because of the historical scenario.

Inputs for calibration

The inputs for the calibration procedure include the autoregressive model, historical demand, historical measurements, initial state vector in deviations $\partial \boldsymbol{x}_0$, covariance matrix of transition error $\boldsymbol{Q}_h$, covariance matrix of measurement error $\boldsymbol{R}_h$ and initial covariance matrix for state vector $\boldsymbol{P}_0$.

An autoregressive process of degree 1 is adopted based on preliminary tests. The generation of historical demand was discussed in the previous section. The historical measurements are the measurements resulting from the historical demand. To calculate historical measurements, we ran DynaMIT with the historical demand 5 times and averaged the results to account for stochasticity in the simulator.

The state vector $\partial \boldsymbol{x}_0$ is set to zero as it represents the deviation from historical values. The covariance matrices $\boldsymbol{Q}_h$ and $\boldsymbol{R}_h$ are constructed assuming that the random errors (elements of $\boldsymbol{v}_h$) are independent of each other. This assumption has been made, because estimating covariances is data-intensive. It requires OD flow estimates for a number of days, where each day forms a single observation in the estimation procedure [4]. Specifically, the diagonal elements of $\boldsymbol{Q}_h$ which represent variance of $\boldsymbol{w}_h$ is set as

$$\boldsymbol{Q}_h = \begin{bmatrix} \max\{q_0, \alpha|\partial x_{h,1}|\}^2 & 0 & \dots & 0 \\ 0 & \max\{q_0, \alpha|\partial x_{h,2}|\}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \max\{q_0, \alpha|\partial x_{h,n}|\}^2 \end{bmatrix} \tag{32}$$

where, $\partial x_{h,j}$ is the j-th element of random vector $\partial \boldsymbol{x}_h$; α is a fraction to tune. The diagonal elements for $\boldsymbol{Q}_h$ are set such that the standard deviation of $w_{h,j}$ is α times the magnitude of $\partial x_{h,j}$. In order to handle the situation where $\partial \boldsymbol{x}_h$ has near zero values, the standard deviation of random variable $w_{h,j}$ is set to $\max\{q_0, \alpha|\partial \boldsymbol{x}_{h,j}|\}$. In our case, α is set to 0.3, q_0 is set to 1, allowing elements in $\partial \boldsymbol{x}_h$ with 0 values to change during the online calibration procedure.

Similarly, the elements of covariance $\boldsymbol{R}_h$, which represent the variance of $\boldsymbol{v}_h$ are set as

$$\boldsymbol{R}_h = \begin{bmatrix} \max\{r_0, \beta|M_{h,1}|\}^2 & 0 & \dots & 0 \\ 0 & \max\{r_0, \beta|M_{h,2}|\}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \max\{r_0, \beta|M_{h,n}|\}^2 \end{bmatrix} \tag{33}$$

The fraction β is chosen to be 0.1, meaning standard deviation is 10% of the sensor readings. r_0 is set to 10, considering the magnitude of sensor readings. Note that the imperfection of local linearization in Equation (2) is also captured in $\boldsymbol{R}_h$.

Finally, $\boldsymbol{P}_0$ is initialized as $\boldsymbol{Q}_0$. It follows from the initialization of $\partial \boldsymbol{x}_0 = \boldsymbol{0}$, as $\boldsymbol{P}_0$ is a diagonal matrix with q_0.

Results and Discussion

In order to quantify the effectiveness of the calibration process in replicating the observed measurements, we used the Root Mean Square Normalized error (RMSN) which is defined as

$$RMSN = 100 \times \frac{\sqrt{N \sum_{j=1}^{N} (\hat{M}_j - M_j)^2}}{\sum_{j=1}^{N} (M_j)} \% \tag{34}$$

where, M_j is the j-th observed (true) measurement value and $\hat{M}_j$ is the j-th simulated (estimated) measurement value. N is the number of sensors. The RMSNs are calculated both interval-wise and for the complete simulation period.

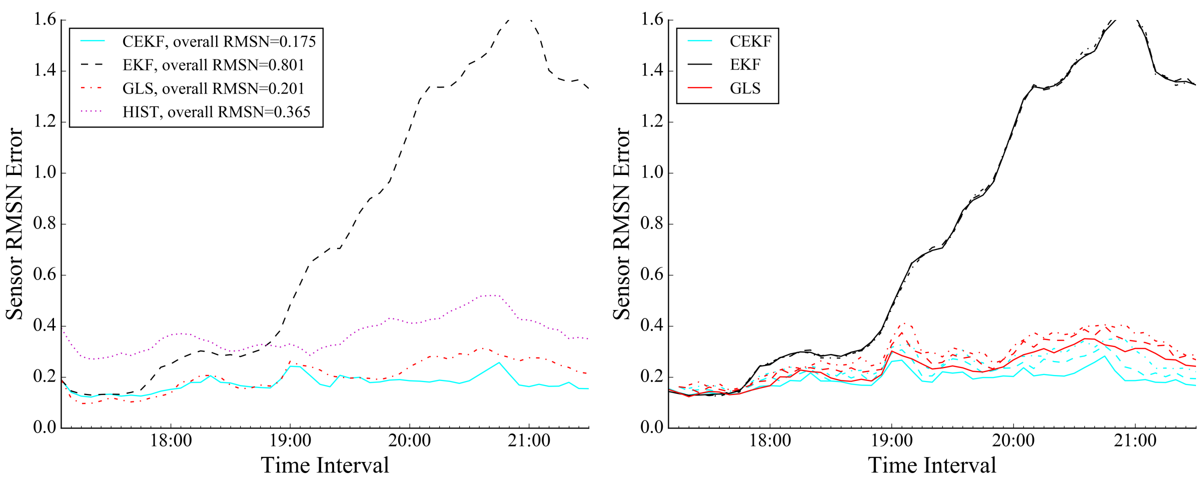
TABLE 2 : Overall Algorithm Performance

(a) RMSN for State Estimation and Prediction

Algorithm	Estimation RMSN	Prediction RMSN		
		Step 1	Step 2	Step 3
Historical	36.50%	36.43%	36.53%	36.66%
EKF	80.08%	80.05%	80.84%	81.92%
GLS	20.05%	22.82%	25.85%	28.29%
CEKF	17.48%	18.91%	21.86%	24.54%

(b) CEKF's Improvement over Other Algorithms for State Estimation and Prediction

Base Algorithm	Estimation Improvement	Prediction Improvement		
		Step 1	Step 2	Step 3
Historical	52.11%	48.09%	40.16%	33.06%
EKF	78.17%	76.38%	72.96%	70.04%
GLS	12.82%	17.13%	15.44%	13.26%

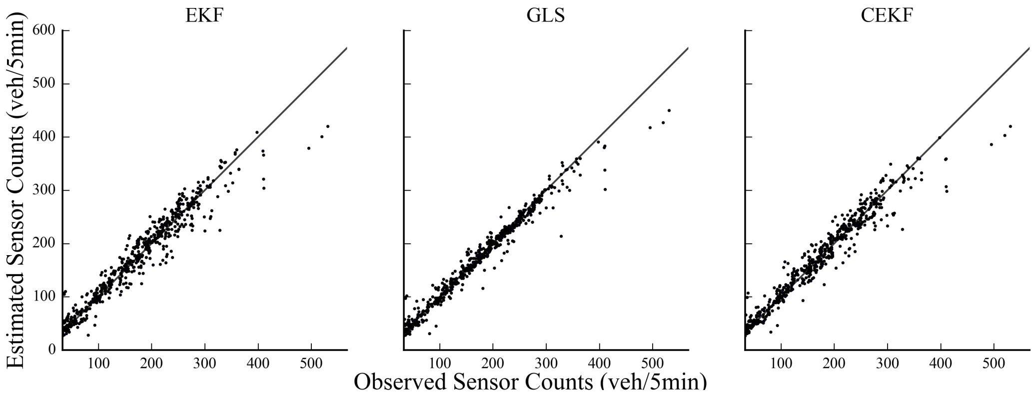


(a) Sensor RMSN in State Estimation (b) Sensor RMSN in 1/2/3-Step State Prediction

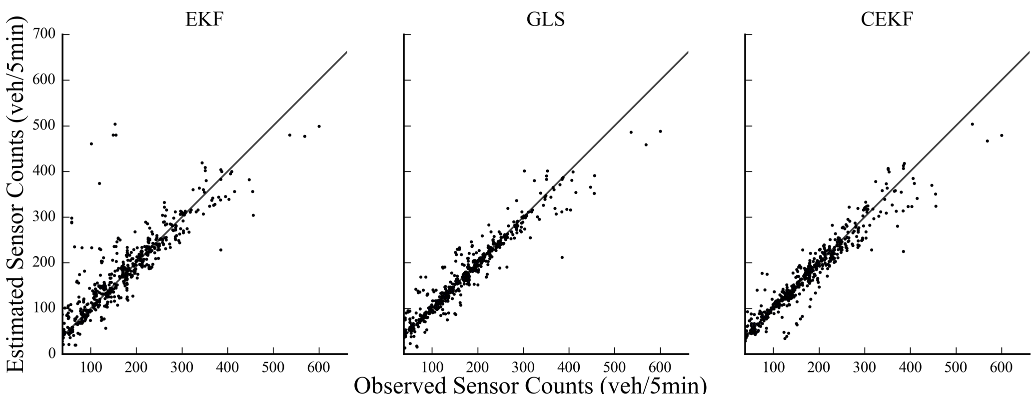
FIGURE 3 : Sensor RMSN in State Estimation and Prediction

The aggregate RMSNs in sensor counts for the entire simulation period are presented in Table 2 along with the percentage improvements of CEKF with respect to the base case (historical or no calibration), EKF and GLS. Figure 3 presents the plots of sensor count RMSNs with respect to time-of-day for each of the methods. The RMSNs in the context of estimation are depicted in Figure 3a and those of prediction in Figure 3b.

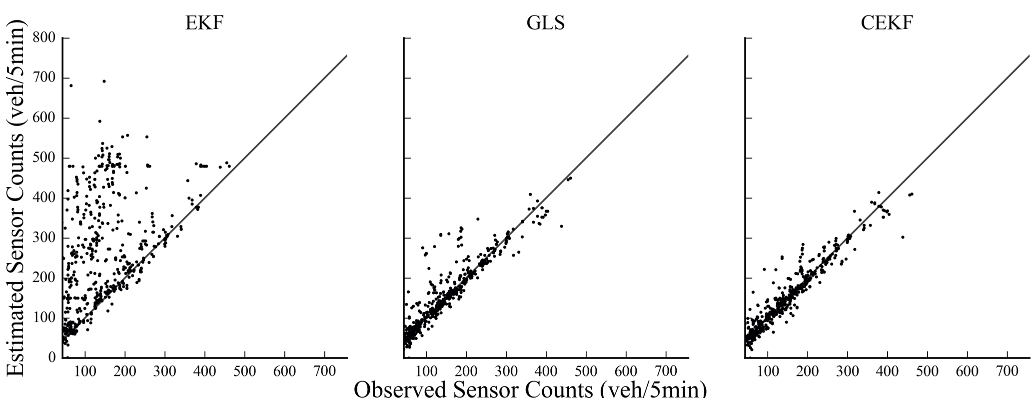
The EKF yields an aggregate RMSN of 80.08% in state estimation, which is significantly worse than without calibration (36.50%). As is shown in Figure 3a, the initial errors for the EKF



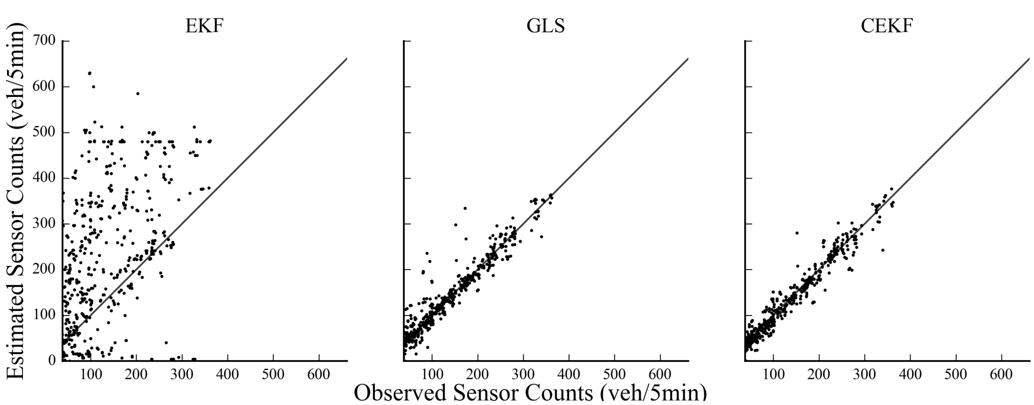
(a) 17:10-17:15



(b) 18:35-18:40



(c) 20:00-20:05



(d) 21:25-21:30

FIGURE 4 : Estimated vs. Observed Flow Counts for EKF, GLS and CEKF in Selected Intervals

are low, but its performance deteriorates with time. Although the divergence appears to abate at around 21:30, its overall performance is still worse than when no calibration is performed.

The GLS approach in contrast (Table 2a) performs well with an aggregate RMSN of 20.05% for estimation and 28.29% for step 3 prediction. The CEKF also performs well with an aggregate RMSN of 17.48% for estimation and 24.54% for 3 step prediction. From Table 2b, the CEKF improves over the historical by 52.11% in estimation and 48.09%, 40.16% and 33.06% in step 1,2 and 3 prediction, respectively. The CEKF also outperforms GLS by 12.82% in estimation and up to 17.13% in prediction.

Figure 3 suggests that the constrained EKF manages to keep the overall *RMSN* at around 18%, and maintain a low *RMSN* until the calibration ends. Note that the oscillation in the first few intervals may be due to an imperfect initial covariance matrix. However, the covariance update process corrects this as the simulation progresses and it outperforms the GLS in the last several intervals.

The previous observations are further corroborated by Figure 4. It presents the scatter plots of estimated vs. observed sensor counts of the three procedures for four estimation intervals. If the sensor counts are estimated exactly, all the points will lie on the 45 degree line. From the plots, the EKF consistently overestimates sensor counts in the later estimation intervals indicating divergence. This, we hypothesize, is the result of the truncation process normally adopted which is discussed in more detail in the subsequent section. On the other hand, the CEKF and GLS appear to estimate the sensor counts consistently well. Again, CEKF's performance improves with time and it eventually performs better than GLS.

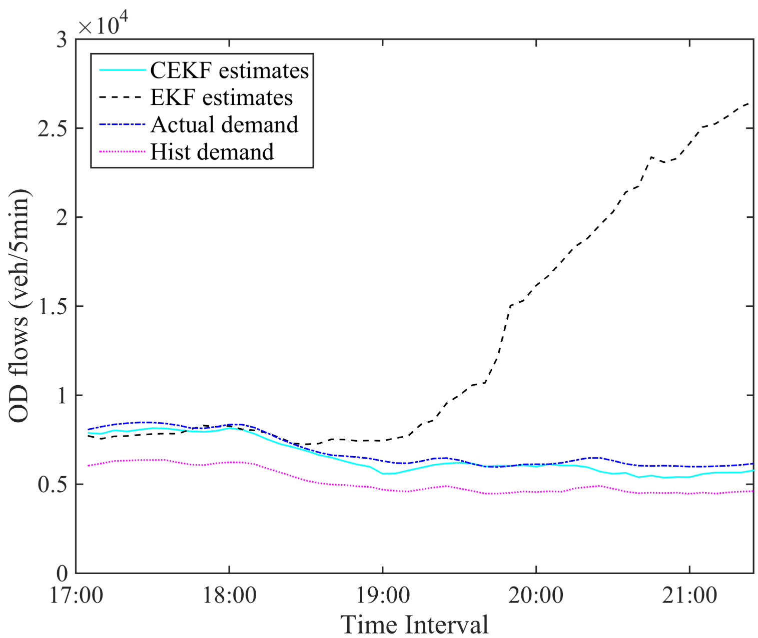


FIGURE 5 : Total Number of Vehicles Departing in each 5-Minute Interval

Divergence of EKF

The results from the earlier section suggest that the EKF diverges. Note that the OD flow estimates from the EKF can be either positive or negative. In the standard EKF, the negative OD flow estimates are truncated to 0 and the non-negative OD flow estimates are kept unchanged. Consequently, the network-level OD flow is over estimated. For example, in a given interval, assume that the network-level flow is estimated to be 10,000 vehicles with the negative OD flow estimates summing up to -1000 and positive flow estimates summing up to 11000. If the negative OD flows are truncated at zero, this in effect will yield a total demand of 11000 and results in 1,000 additional vehicles on the network. In the next interval, as the estimated sensor counts will be higher the EKF will attempt to reduce the net demand by decreasing the total number of vehicles (total OD flows) and setting more OD flows to negative values. The subsequent truncation further exacerbates the problem leading to the poor performance of EKF.

The overestimation of demand is visible in the plot of total number of vehicles estimated in each interval for the EKF (Figure 5). It can be seen that at around 18:30, the total estimated OD flow starts to deviate substantially from the actual demand. This leads to the large errors in both estimation and prediction which also start to increase significantly at around the same time interval (Figure 3). Although some studies [19, 20] have observed the divergence of EKF due to linearization, the results here indicate that improper modeling of constraints may also result in divergence.

In contrast, the proposed CEKF procedure clearly overcomes this issue and even outperforms the GLS method in both state estimation and prediction.

8 CONCLUSION

This paper addressed the problem of calibration of large scale simulation-based Dynamic Traffic Assignment (DTA) models. To overcome a limitation of existing Kalman Filter (KF) based methods, namely the inability to model constraints on the calibration parameters, a new Constrained Extended Kalman Filter (CEKF) method was presented. Given the state estimates and posterior covariance matrix from the KF, the problem of computing the maximum a posteriori (MAP) estimates subject to constraints is formulated as a constrained quadratic program. In addition, a heuristic solution procedure is proposed to solve this quadratic program that yields solutions close the true optimum (around 0.001% worse in objective function value). Further, a coordinate descent algorithm is applied using the heuristic solution as a starting point to optimally solve for the constrained MAP estimates. Numerical tests have shown that the combined algorithm attains the optimum in the same precision as coordinate descent and *quadprog* in MATLAB, but 31.5% and 94.9% faster.

Experiments using the DTA system DynaMIT on the Singapore expressway network indicate that the proposed CEKF has improved the standard EKF by 78.17% in state estimation and up to 76.38% in state prediction. The CEKF also outperforms GLS method by 12.82% in state estimation and up to 17.13% in state prediction. The proposed method has important applications in the offline and online calibration of DTA systems which are essential for obtaining accurate estimates and predictions of traffic states.

Future directions of research include more extensive testing of the CEKF method and improvement of computational efficiency to facilitate deployment in an online setting.

9 ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its CREATE program, Singapore-MIT Alliance for Research and Technology (SMART) Future Urban Mobility (FM) IRG.

The authors would also like to thank Yundi Zhang for his support in the data smoothing

procedure.

REFERENCES

[1] Chiu, Y.-C., J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks, Dynamic traffic assignment: A primer. *Transportation Research E-Circular*, , No. E-C153, 2011.

[2] Antoniou, C., H. N. Koutsopoulos, M. Ben-Akiva, and A. S. Chauhan, Evaluation of diversion strategies using dynamic traffic assignment. *Transportation planning and technology*, Vol. 34, No. 3, 2011, pp. 199–216.

[3] Ben-Akiva, M., H. N. Koutsopoulos, C. Antoniou, and R. Balakrishna, Traffic simulation with DynaMIT. In *Fundamentals of traffic simulation*, Springer, 2010a, pp. 363–398.

[4] Antoniou, C., *On-line calibration for dynamic traffic assignment*. Ph.D. thesis, Massachusetts Institute of Technology, 2004.

[5] Balakrishna, R., *Off-line calibration for dynamic traffic assignment models*. Ph.D. thesis, Massachusetts Institute of Technology, 2006.

[6] Cascetta, E., D. Inaudi, and G. Marquis, Dynamic estimators of origin-destination matrices using traffic counts. *Transportation science*, Vol. 27, No. 4, 1993, pp. 363–373.

[7] Zhou, X. and H. Mahmassani, Online consistency checking and origin-destination demand updating: Recursive approaches with real-time dynamic traffic assignment operator. *Transportation Research Record: Journal of the Transportation Research Board*, , No. 1923, 2005, pp. 218–226.

[8] Balakrishna, R., M. Ben-Akiva, and H. Koutsopoulos, Offline calibration of dynamic traffic assignment: simultaneous demand-and-supply estimation. *Transportation Research Record: Journal of the Transportation Research Board*, 2007.

[9] Ashok, K., *Estimation and prediction of time-dependent origin-destination flows*. Ph.D. thesis, Massachusetts Institute of Technology, 1996.

[10] Ashok, K. and M. E. Ben-Akiva, Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. *Transportation Science*, Vol. 36, No. 2, 2002, pp. 184–198.

[11] Zhou, X. and H. S. Mahmassani, A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework. *Transportation Research Part B: Methodological*, Vol. 41, No. 8, 2007, pp. 823–840.

[12] Antoniou, C., M. Ben-Akiva, and H. N. Koutsopoulos, Nonlinear Kalman filtering algorithms for on-line calibration of dynamic traffic assignment models. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 8, No. 4, 2007, pp. 661–670.

[13] Antoniou, C., M. Ben-Akiva, and H. N. Koutsopoulos, Dynamic traffic demand prediction using conventional and emerging data sources. In *IEE Proceedings-Intelligent Transport Systems*, IET, 2006, Vol. 153, pp. 97–104.

[14] Simon, D. and D. L. Simon, Kalman filtering with inequality constraints for turbofan engine health estimation. In *Control Theory and Applications, IEE Proceedings-*, IET, 2006, Vol. 153, pp. 371–378.

[15] Ungarala, S., E. Dolence, and K. Li, Constrained extended Kalman filter for nonlinear state estimation. *IFAC Proceedings Volumes*, Vol. 40, No. 5, 2010, pp. 63–68.

[16] Zhang, H., *Constrained Extended Kalman Filter: an Efficient Improvement of Calibration for Dynamic Trac Assignment Models*. Master's thesis, Massachusetts Institute of Technology, 2016.

[17] Lu, L., *W-SPSA: an Efficient Stochastic Approximation Algorithm for the off-line calibration of Dynamic Traffic Assignment models*. Master's thesis, Massachusetts Institute of Technology, 2013.

[18] Ben-Akiva, M., H. N. Koutsopoulos, T. Toledo, Q. Yang, C. F. Choudhury, C. Antoniou, and R. Balakrishna, Traffic simulation with MITSIMLab. In *Fundamentals of Traffic Simulation*, Springer, 2010b, pp. 233–268.

[19] Bizup, D. F. and D. E. Brown, The over-extended Kalman filter-don't use it! In *Proceedings of the Sixth International Conference of Information Fusion*, 2003, Vol. 1, pp. 40–46.

[20] Perea, L., J. How, L. Breger, and P. Elosegui, Nonlinearity in sensor fusion: Divergence issues in EKF, modified truncated SOF, and UKF, 2007.